Information Sciences 255 (2014) 30–44

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Approximating the optimal mapping for weapon target assignment by fuzzy reasoning

Mehmet Alper Şahin[a,*], Kemal Leblebicioğlu[b]

[a] HAVELSAN, Ankara, Turkey

[b] Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 25 November 2011
Received in revised form 25 May 2013
Accepted 5 August 2013
Available online 12 August 2013

Keywords:
Weapon Target Assignment (WTA)
Fuzzy Decision Maker (FDM)
Grid partitioning

ABSTRACT

Weapon target assignment is a weapon assignment problem with the objective of minimizing the expected survival value of targets. This problem must be quickly solved on the battlefield (i.e., in real-time). Considering the combined complexity and the strict time constraints, a fuzzy decision maker is proposed as an alternative to aid commanders in deciding on proper weapon assignments. The concept builds a fuzzy decision maker for a given data set by using extended grid partitioning based on a sensitivity analysis of input variables. The proposed decision maker is implemented and tested with several randomly sampled assignment instances on realistic scenarios. In addition, a larger-scale scenario is also considered to demonstrate the scalability of the approach. The results show that the proposed system exhibits satisfactory performance and is serviceable on the battlefield.

1. Introduction

Weapon Target Assignment (WTA) refers to the reactive assignment of defensive weapons to engage or counter identified threats. As an illustrative example, consider the air defense of a naval battle group. The assets being defended are aircraft carriers, escort warships and support ships, which all have some intrinsic value to the defense. The threats are enemy anti-ship missiles launched from submarines, surface ships and air platforms. The weapons are any measure that can be used to eliminate the threats, e.g., guided missiles, fast firing guns, etc. The problem is to assign the proper weapon to each threat. The primary concern is minimizing the total expected lethality of the enemy missiles with a limited number of weapons. This problem is known as WTA in Operations Research; interested readers can refer to [8,13,15,23,26] for a comprehensive literature review of WTA.

Weapons are assigned to threats in stages based on the observed outcomes of previous stages [15]. There is limited time to observe, decide and act; the problem must be solved in a short period of time. For example, a naval battle group can detect an enemy anti-ship missile at 10 km away; this allows for (at most) 10 s to counter the anti-ship missile. It is therefore crucial to act as close to real-time as possible, provided that all acts are consistent with the mission objectives and compliant with platforms and environmental constraints.

Various solvers have been developed for WTA. There are global optimum solvers, such as cutting plane techniques and branch and bound algorithms. Ahuja et al. [1,2] proposed several branch and bound algorithms with various bounding strategies for WTA. In addition, several exact solvers have been proposed with some restrictive conditions, i.e., when

* Corresponding author. Tel.: +90 5300410207.
E-mail address: m.alper.sahin@hotmail.com (M.A. Şahin).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.08.004

all the weapons are identical [12,16] or when targets can receive at most one weapon [9]. There are also several global search solvers, such as genetic algorithms [3,10,20,36], simulated annealings [7], discrete particle swarm optimizations [35], permutation and tabu search heuristics [32], and a hybrid algorithm of a genetic algorithm with an ant colony optimization [19].

WTA is known to be a typical Non-Polynomial Complete (NPC) problem [21] such that any enumeration-based solver faces exponential computational complexity as the problem size increases. In other words, while any enumeration method may ensure a global solution, its computational requirement grows exponentially as the number of targets and/or weapons grows. Problems on the battlefield must be solved quickly, i.e., close to real-time. The combinatorial complexity of the problems implies that, even with the supercomputers available today, optimal solutions cannot be obtained close to real-time, leaving insufficient time to act. However, while global search solvers are reported to be efficient to some extent, none of them can guarantee optimal results. More importantly, they cannot guarantee the competence of a result obtained after a certain run time. To address the computational burden of exact solvers and the search times of global search solvers, several sophisticated search algorithms and heuristics have been recently proposed. Rosenberger et al. [25] and Bogdanowicz [5] applied an auction algorithm for WTA. Lee [18] presented an enhanced large-scale neighborhood (VLSN) search algorithm for WTA. Xin et al. [31] proposed a rule-based heuristic using the domain knowledge of WTA. They reported a very efficient computation time, even for large-scale WTA, but they could not ensure a global solution; the auction-based solvers only guarantee optimality in one-to-one assignment cases. However, VLSN search algorithms and rule-based heuristics can return a feasible solution (a solution satisfying all constraints) of the highest possible quality. Interested readers can refer to [6] for a general overview of the meta-heuristics of hard problems. However, none of the aforementioned solvers satisfies optimality on a strict time constraint.

Rule-based expert systems are increasingly preferred over controlled complex real-time systems. The complexity and the strict time constraint suggest a rule-based expert system as an alternative solver. Rule-based mapping is known to be efficient in computation time by its nature. However, rule-based systems may appear to have an infeasible number of rules for such complex problems. The approximate reasoning theory [33], along with the compositional rule of inference method, provides a useful framework for constructing a rule-based solution with a reasonable number of rules. Observations on a battlefield are characterized by imprecision and uncertainty. Based on the problem complexity, the strict time constraints, the number of rules, and the imprecision and uncertainty of observations, we propose to apply the approximate reasoning theory to construct a rule-based solver as an alternative decision aid system for helping battlefield commanders decide on proper WTA. In our previous work [27,28], we employed the idea of applying the approximate reasoning theory for WTA. However, we could not analyze the theoretical aspects and the results were insufficient. In this work, we can better justify the concept by including an analysis of the approximation properties.

The primary goal of this work is to show the applicability of the approximate reasoning theory for WTA. The challenge is to extract assignment rules and generate a rule-base. We utilize a grid partitioning approach [14] in this work and extend it in such a way that regions having a large effect on outputs are partitioned into smaller grids. To show the principle of the approach, we concentrate on small-scale WTA, i.e., a single platform perspective; these small-scale problems include a small number of weapons and targets. For example, [4,32] defined single platform scenarios with only five weapons and three targets. Simulated results showed the superiority of the proposed model at a small-scale WTA. However, WTA can be considered from several different perspectives from single platform combat to force coordination [24]. In the force perspective, the number of weapons may easily reach 100 or more. Ahuja et al. [1,2] classified such force perspective problems as large-sized and fairly large-sized. To demonstrate the scalability of the approach, a large-scale scenario is also considered in this work. The results show that the proposed system exhibits satisfactory performance and is serviceable on a battlefield.

Section 2 introduces a formal formulation of WTA. Section 3 presents the fuzzy decision maker. Section 4 discusses the selection of the grid counts and the grid partitioning. Section 5 analyzes the approximation properties. Section 6 presents the simulation results; several existing solutions were also employed for comparison. The proposed decision aid system outperforms its competitors on all test problems. Finally, Section 7 concludes the paper.

2. Problem formulation: Weapon Target Assignment (WTA)

Research on WTA dates back to the 1950s, where WTA modeling issues were first investigated [11,22]; since then, the mathematical models have significantly improved [1,2,20,21,35]. The problem is formulated as a non-linear integer programming problem, with the objective of minimizing the total lethality of the targets. The problem variables are the kill probabilities of the weapon-target pairs and the lethality values of the targets. The kill probability is the weapon's probability of destroying the target if engaged. It depends on all aspects of engagement, such as the type of weapon and the type, state, and location (range, sector) of the target. The lethality value indicates how lethal the target is to the asset if it survives to its destination. This depends on which asset is at risk, as well as the type, location (range, sector), velocity, and course of the target. This formulation allows for a preferential defense under a heavy attack; it may be optimal to leave the targets aimed at low-value assets undefended and concentrate on destroying the targets aimed at high-value assets.

Consider the assignment problem of w weapons to t targets: Let $p_{ij} \in [0,1]$ be the kill probability of weapon j for target i and $v_i \in [0,1]$ be the survival value of target i where $i = 1, \ldots, t$ and $j = 1, \ldots, w$. The WTA is formulated as a non-linear integer programming problem as follows:

minimize

$$\sum_{i=1}^{t} v_i \prod_{j=1}^{w} (1 - p_{ij})^{d_{ij}} \tag{1a}$$

subject to

$$\sum_{i=1}^{t} d_{ij} \leqslant 1, \quad \forall\, j = 1, \ldots, w \tag{1b}$$

$$d_{ij} \in \{0, 1\}, \quad \forall\, i = 1, \ldots, t \quad , \forall\, j = 1, \ldots, w \tag{1c}$$

where d_{ij} is the decision value indicating if weapon j is assigned to target i. In the above formulation (1), we minimize the sum of the weighted cumulative probability of threat lethality while ensuring that a weapon can be assigned to (at most) one target. We may consider adding additional constraints, such as a lower bound on the reduction in target lethality value or a lower and upper bound on the number of weapons assigned to a target.

3. Fuzzy Decision Maker (FDM)

Consider the assignment of w weapons to t threats. Let us then collect all problem variables, i.e., v_i and p_{ij} where $i = 1, \ldots, t$ and $j = 1, \ldots, w$, in an $n = t + tw$ dimensional vector as follows:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \tag{2a}$$

$$= (v_1, \ldots, v_t, p_{11}, \ldots, p_{t1}, \ldots, p_{tw}) \in I \tag{2b}$$

where I is the set of all input vectors, i.e., $I = \{(x_1, x_2, \ldots, x_n) | x_i \in [0,1], \forall\, i = 1, \ldots, n\}$. Similarly, let us also collect all decision values, i.e., d_{ij}, where $i = 1, \ldots, t$ and $j = 1, \ldots, w$, in an $m = tw$ dimensional vector as follows:

$$\mathbf{y} = (y_1, y_2, \ldots, y_m) \tag{3a}$$

$$= (d_{11}, \ldots, d_{t1}, d_{12}, \ldots, d_{t2}, \ldots, d_{tw}) \in O \tag{3b}$$

where O is the decision vector set, i.e., $O = \{(y_1, y_2, \ldots, y_m) | y_i \in \{0,1\}, \forall\, i = 1, \ldots, m\}$.

The decision vector set is actually a finite set with $c = 2^m$ possible assignment decisions. Thus, we may rewrite O as $O = \{class_1, class_2, \ldots, class_c\}$, where *class* is any possible assignment decision. For example, $class_1 = (0, 0, 0, \ldots, 0)_{1\times m}$ means no assignment, $class_2 = (1, 0, 0, \ldots, 0)_{1\ \times m}$ means that weapon $j = 1$ is assigned to only threat $i = 1$ and all other weapons are not assigned. In this context, the WTA can be modeled as a classification problem where the input pattern, i.e., $\mathbf{x} \in I$, is assigned to one of the given set of decisions, i.e., $\mathbf{y} \in O = \{class_1, class_2, \ldots, class_c\}$.

The discussion above shows that the WTA can be modeled as a classification problem. Therefore, a single fuzzy classifier (e.g., a Mamdani-type classifier) can be used to solve the problem. However, we must handle a considerable number of classes, resulting in an impractical rule-base size, even for a small-scale assignment problem. For example, the total class count would be $c = 2^{25} = 33{,}554{,}432$ for five weapons and five targets. An obvious solution is to remodel the problem as a union of individual classification problems.

Based on the given rationale, we divide set O into w individual subsets as $O = O_1 \cup \ldots \cup O_j \cup \ldots \cup O_w$ such that each holds the possible assignment of only one weapon. That is, $O_j = \left\{class_0^j, \ldots, class_i^j, \ldots, class_t^j\right\}$ is the set holding the possible assignments of weapon $j = 1, 2, \ldots, w$ where $class_0^j = (0, 0, \ldots, 0)_{1\times t}$ means no assignment and $class_i^j = (0, 0, \ldots, 1, \ldots 0)_{1\times t}$ indicates that weapon j is assigned to threat i. Thus, the FDM can be built as a union of w individual Mamdani-type classifiers. Moreover, the cascaded structure shown in Fig. 1 can reduce the number of inputs for each classifier. Therefore, it is sufficient to handle only $t + 1$ classes and t inputs within a single classifier, obviously decreasing the complexity.

As shown in Fig. 1, the FDM consists of two units: the Fuzzy Transformation Unit (FTU) and the Fuzzy Classifier Unit (FCU). The first unit (FTU) approximates the data set to an intermediate domain where assignment decisions are not necessarily integers. The second unit (FCU) classifies the FTU output to a valid assignment decision. The FTU actually performs a non-linear transformation of the original input pattern to an intermediate pattern. The transformation is expected to increase the discrimination performance of the FCU, leading to a more accurate decision. Moreover, the FCU receives a subset of the intermediate pattern (instead of the entire pattern) without regard to informational loss. Therefore, the FCU can be constructed with a less complex rule-base. In the following, we proceed with the details of each unit.

3.1. Fuzzy Transformation Unit (FTU)

As depicted in Fig. 1, (2) is the FTU input. The goal is to approximate this input to an intermediate pattern denoted by vector $\mathbf{u} = (u_1, u_2, \ldots, u_m)$. Let U be the set of all possible intermediate patterns. That is, $U = \{(u_1, u_2, \ldots, u_m) | u_i \in [0,1];\ \forall\, i = 1, \ldots, m\}$. Note that u_i does not necessarily constitute a valid decision.

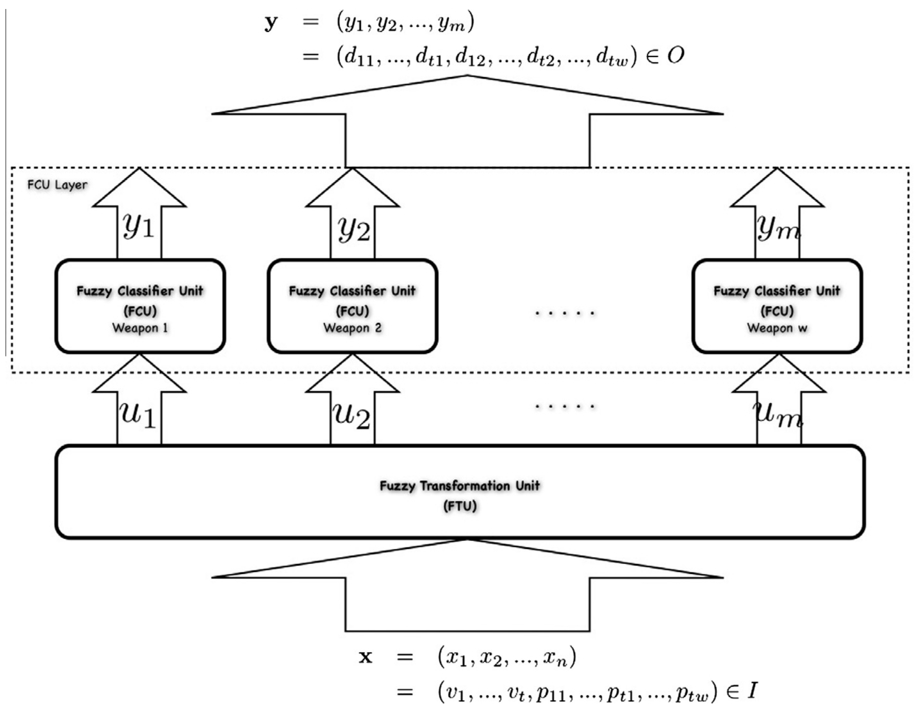
$\mathbf{y} = (y_1, y_2, ..., y_m) = (d_{11}, ..., d_{t1}, d_{12}, ..., d_{t2}, ..., d_{tw}) \in O$
FCU Layer
y_1
y_2
y_m
Fuzzy Classifier Unit (FCU) Weapon 1
Fuzzy Classifier Unit (FCU) Weapon 2
Fuzzy Classifier Unit (FCU) Weapon w
u_1
u_2
u_m
Fuzzy Transformation Unit (FTU)
$\mathbf{x} = (x_1, x_2, ..., x_n) = (v_1, ..., v_t, p_{11}, ..., p_{t1}, ..., p_{tw}) \in I$

Fig. 1. Structure of the Fuzzy Decision Maker (FDM).

Suppose that the domain interval of each input variable is partitioned into h_i fuzzy sets such as $A^i_{j_i}$ where $i = 1, \ldots, n$ and $j_i = 1, \ldots, h_i$. Let $A_{j_1 j_2,\ldots,j_n}$ be the fuzzy set defining the rule premise, i.e., $A_{j_1 j_2,\ldots,j_n} = A^1_{j_1} \times A^2_{j_2} \times \ldots \times A^n_{j_n}$. The FTU is then characterized by the following rule set.

$$\text{RULE}_{j_1 j_2,\ldots,j_n} : \text{IF } \mathbf{x} \in A_{j_1 j_2,\ldots,j_n} \text{ THEN } \mathbf{u} \text{ is } \mathbf{B_{j_1 j_2,\ldots,j_n}} \tag{4}$$

where $\mathbf{B_{j_1 j_2,\ldots,j_n}} \in U$ is the rule output. Let $\mu_{A^i_{j_i}}(.)$ be the membership function corresponding to fuzzy set $A^i_{j_i}$. The $\mathbf{B_{j_1 j_2,\ldots,j_n}}$ is then evaluated by

$$\mathbf{B_{j_1 j_2,\ldots,j_n}} = \frac{\sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \mu_{A_{j_1 j_2,\ldots,j_n}}(\bar{\mathbf{x}})\bar{\mathbf{y}}}{\sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \mu_{A_{j_1 j_2,\ldots,j_n}}(\bar{\mathbf{x}})} \tag{5}$$

where $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle$ is any data tuple in the training data set S and $\mu_{A_{j_1 j_2,\ldots,j_n}}(\mathbf{x}) = \prod_{i=1}^{n} \mu_{A^i_{j_i}}(x_i)$.

Once the rule-base is formed, we may represent the FTU as a Takagi–Sugeno (TS) model [30].

$$\mathbf{u} = \frac{\sum_{j_1 j_2,\ldots,j_n \in H} \mu_{A_{j_1 j_2,\ldots,j_n}}(\mathbf{x}) \mathbf{B_{j_1 j_2,\ldots,j_n}}}{\sum_{j_1 j_2,\ldots,j_n \in H} \mu_{A_{j_1 j_2,\ldots,j_n}}(\mathbf{x})} \tag{6}$$

where $H = \{j_1, j_2, \ldots, j_n | j_1 = 1, 2, \ldots, h_1;\ j_2 = 1, 2, \ldots, h_2, \ldots, j_n = 1, 2, \ldots, h_n\}$. Note that the TS-type fuzzy system does not require a separate defuzzifier.

To further simplify the model (6), we define

$$\phi_{j_1 j_2,\ldots,j_n}(\mathbf{x}) = \frac{\mu_{A_{j_1 j_2,\ldots,j_n}}(\mathbf{x})}{\sum_{j_1 j_2,\ldots,j_n \in H} \mu_{A_{j_1 j_2,\ldots,j_n}}(\mathbf{x})} \tag{7}$$

$$\varphi_{j_1 j_2,\ldots,j_n}(\mathbf{x}) = \frac{\mu_{A_{j_1 j_2,\ldots,j_n}}(\mathbf{x})}{\sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \mu_{A_{j_1 j_2,\ldots,j_n}}(\bar{\mathbf{x}})} \tag{8}$$

Using the above notations, the FTU is finally expressed as follows:

$$\mathbf{u} = FTU(\mathbf{x}) \tag{9a}$$

$$= \sum_{j_1 j_2,\ldots,j_n \in H} \phi_{j_1 j_2,\ldots,j_n}(\mathbf{x}) \mathbf{B_{j_1,\ldots,j_n}} \tag{9b}$$

where

$$\mathbf{B}_{\mathbf{j_1 j_2,\ldots,j_n}} = \sum_{\langle \bar{\mathbf{x}},\bar{\mathbf{y}}\rangle \in S} \varphi_{j_1 j_2,\ldots,j_n}(\bar{\mathbf{x}})\bar{\mathbf{y}} \tag{10}$$

3.2. Fuzzy Classifier Unit (FCU)

As depicted in Fig. 1, there exist w FCUs. To reduce the rule-base complexity, we relate each FCU with only one weapon. That is, each FCU classifies (maps) a part of the intermediate pattern (i.e., FTU output) to an assignment decision for only one weapon. For example, $\mathbf{u_j} = (u_{(j-1)t+1}, u_{(j-1)t+2}, \ldots, u_{(j-1)t+t}) \subset U$ is the input of the FCU related to weapon j where $j = 1, \ldots, w$. Note that the total number of input variables is only t. For this example, the possible assignment decisions are $O_j = \left\{class_0^j, \ldots, class_i^j, \ldots, class_t^j\right\} \subset O$, where $class_0^j = (0, 0, \ldots, 0)_{1\times t}$ means no assignment and $class_i^j = (0, 0, \ldots, 1, \ldots 0)_{1\times t}$ indicates that weapon j is assigned to target i.

Suppose that the domain interval of each input variable is partitioned into g_i fuzzy sets: $C_{j_i}^i$, where $i = 1, \ldots, m$ and $j_i = 1, \ldots, g_i$. Thus, for weapon $j = 1, \ldots, w$, the FCU is characterized by the following rule set:

$$\text{RULE}_{j_1 j_2,\ldots,j_t} : \text{IF } \mathbf{u_j} \in C_{j_1 j_2,\ldots,j_t} \text{ THEN } \mathbf{y_j} \text{ is } \mathbf{D}_{\mathbf{j_1 j_2,\ldots,j_t}} \tag{11}$$

where $C_{j_1 j_2,\ldots,j_t}$ is the fuzzy set defining the rule premise. That is, $C_{j_1 j_2,\ldots,j_t} = C_{j_{(j-1)t+1}}^{(j-1)t+1} \times C_{j_{(j-1)t+2}}^{(j-1)t+2} \times \ldots \times C_{j_{jt}}^{jt}$. Moreover, $\mathbf{D}_{\mathbf{j_1 j_2,\ldots,j_t}}$ is rule output and is the common decision encountered in set

$$S_{j_1,\ldots,j_t} = \{(\bar{y}_{(j-1)t+1}, \bar{y}_{(j-1)t+2}, \ldots, \bar{y}_{jt}) | \langle \bar{\mathbf{u}}, \bar{\mathbf{y}}\rangle \in S', \bar{\mathbf{u}}_\beta \in C_{j_1,\ldots,j_t}\} \tag{12}$$

where S' is the training data set of the FCUs. S' is obtained from the training data set S as follows:

$$S' = \{\langle \bar{\mathbf{u}}, \bar{\mathbf{y}}\rangle | \langle \bar{\mathbf{x}}, \bar{\mathbf{y}}\rangle \in S, \bar{\mathbf{u}} = FTU(\bar{\mathbf{x}})\} \tag{13}$$

Once the rule-base is formed, it is easy to obtain the corresponding fuzzy classifier. Let $\mu_{C_{j_i}^i}(.)$ be the membership function corresponding to fuzzy set $C_{j_i}^i$. The rule firing strength for the FCU of weapon $j = 1, \ldots, w$, is then computed by the following equation:

$$\phi_{j_1 j_2,\ldots,j_t}(\mathbf{u_j}) = \frac{\mu_{C_{j_1 j_2,\ldots,j_t}}(\mathbf{u_j})}{\sum_{j_1 j_2,\ldots,j_t \in G_j} \mu_{C_{j_1 j_2,\ldots,j_t}}(\mathbf{u_j})} \tag{14}$$

where

$$\mu_{C_{j_1 j_2,\ldots,j_t}}(\mathbf{u_j}) = \prod_{i=1}^{t} \mu_{C_{j_{(j-1)t+i}}^{(j-1)t+i}}(u_{(j-1)t+i}) \tag{15}$$

and

$$G_j = \{(j_{(j-1)t+1}, j_{(j-1)t+2}, \ldots, j_{jt}) | j_{(j-1)t+1} = 1, 2, \ldots, g_{(j-1)t+1}; \ldots; j_{jt} = 1, 2 \ldots, g_{jt}\} \tag{16}$$

Once the firing strengths for weapon $j = 1, \ldots, w$ are determined, the output of the FCU is evaluated by

$$\mathbf{y_j} = \mathbf{D}_{\mathbf{j_1^* j_2^*,\ldots,j_t^*}} \tag{17}$$

where

$$j_1^*, j_2^*, \ldots, j_t^* = \arg\max_{j_1,\ldots,j_t \in G_j} (\phi_{j_1 j_2,\ldots,j_t}(\mathbf{u_j})) \tag{18}$$

4. Training

To apply grid partitioning, we must properly set grid counts (h_i and g_j) while training both the FTU and the FCU. Obviously, the choice of grid counts has a significant influence on the performance of the trained rule-base. If the count is too small, the rule-base will be insufficient for modeling non-linear behavior. However, if it is too large, the corresponding rule-base will be too complex to implement.

Suppose that we have an upper bound on the rule count and a corresponding upper bound on the number of grids. The input variables that have a high influence on output should obviously be partitioned with the smallest possible grids. Based on this rationale, the rule-base can achieve satisfactory performance, even with a lower bound on the rule count. In this section, we propose an extended grid partitioning methodology that uses a sensitivity level for each input variable to set grid counts.

4.1. Sensitivity analysis of input variables

Let $[y_1, y_2, \ldots, y_m] = f(x_1, x_2, \ldots, x_n)$ be a multi-input/multi-output non-linear function. Let $\{\langle \mathbf{x_1}, \mathbf{y_1}\rangle, \langle \mathbf{x_2}, \mathbf{y_2}\rangle, \ldots, \langle \mathbf{x_q}, \mathbf{y_q}\rangle\}$ be a set of q data derived from function f. In statistics, a non-linear regression method suggests approximating a non-linear function with a linear one [29]. Thus, we may approximate the derivative of the input variables K as follows:

$$\Delta Y = K \Delta X \tag{19}$$

where

$$\Delta Y = \begin{bmatrix} \bar{\mathbf{y}}_\mathbf{1} - \bar{\mathbf{y}}_\mathbf{2} \\ \bar{\mathbf{y}}_\mathbf{1} - \bar{\mathbf{y}}_\mathbf{3} \\ \vdots \\ \bar{\mathbf{y}}_\mathbf{i} - \bar{\mathbf{y}}_\mathbf{j} \\ \vdots \\ \bar{\mathbf{y}}_\mathbf{q-1} - \bar{\mathbf{y}}_\mathbf{q} \end{bmatrix} \tag{20}$$

and

$$\Delta X = \begin{bmatrix} \bar{\mathbf{x}}_\mathbf{1} - \bar{\mathbf{x}}_\mathbf{2} \\ \bar{\mathbf{x}}_\mathbf{1} - \bar{\mathbf{x}}_\mathbf{3} \\ \vdots \\ \bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j} \\ \vdots \\ \bar{\mathbf{x}}_\mathbf{q-1} - \bar{\mathbf{x}}_\mathbf{q} \end{bmatrix} \tag{21}$$

Note that in cases where there is no exact solution of (19), we may approximate K by following the pseudo-inverse formula:

$$K = (\Delta X^T \Delta X)^{-1} \Delta X^T \Delta Y \tag{22}$$

Each element of K (19) implies an approximate rate of change in output related to the rate of change in input. In other words, K holds the sensitivity level for each input variable. Thus, we may evaluate the sensitivity level for any input variable x_i where $i = 1, \ldots, n$ as follows:

$$l_i = \frac{\| k_{i1} \quad k_{i2} \quad \ldots \quad k_{im} \|}{\sum_{p=1}^{n} \| k_{p1} \quad k_{p2} \quad \ldots \quad k_{pm} \|} \tag{23}$$

where $\|\cdot\|$ is the L-norm ($L = 2$) and $[\, k_{i1} \quad k_{i2} \quad \ldots \quad k_{im} \,]$ is the ith row of the matrix K.

4.2. Rule extraction based on sensitivity analysis

Based on the sensitivity analysis just described, the sensitivity level for each input variable of an FTU can be obtained, i.e., l_i for each x_i where $i = 1, \ldots, n$. We first form the input variation matrix and the output variation matrix on the training data set S as in (20) and (21). Once we determine ΔX and ΔY, we can easily evaluate the sensitivity level by (23) for each input variable.

Suppose that we have an upper bound (*MaxRuleCount*) on the rule count for the FTU. In grid partitioning, the rule count is equal to the multiplication of the grid counts.

$$RuleCount = \prod_{i=1}^{n} h_i \tag{24}$$

Using the result of the sensitivity analysis, we can identify grid count h_i as follows:

$$h_i = \left\lfloor l_i \sqrt[n]{\frac{MaxRuleCount}{\prod_{j=1}^{n} l_j}} \right\rfloor \tag{25}$$

for each x_i where $i = 1, \ldots, n$.

As for FCUs, we can form an input variation matrix for the training data set S' (see (13)) as follows:

$$\Delta U = \begin{bmatrix} \bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2 \\ \bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_3 \\ \vdots \\ \bar{\mathbf{u}}_i - \bar{\mathbf{u}}_j \\ \vdots \\ \bar{\mathbf{u}}_{q-1} - \bar{\mathbf{u}}_q \end{bmatrix} \tag{26}$$

Note that the output variation matrix ΔY is already formed above.

Similar to the identification of the grid count for the FTU, we identify the grid count g_i for the FCU as follows:

$$g_i = \left\lfloor l_i \sqrt[t]{\frac{MaxRuleCount}{\sum_{j=1}^{w}\left[\prod_{p=(j-1)t+1}^{jt} l_p\right]}} \right\rfloor \tag{27}$$

where *MaxRuleCount* is the upper bound on the total rule count for the FCU layer.

By adopting the described methodology, we can properly set grid counts for a given upper bound on a total rule count. Moreover, rule-bases will be more specific for input variables, thereby having a significant effect on output yet general enough to cover variations for all input variables. Thus, we may implement more accurate fuzzy decision makers, even with limited hardware (i.e., a bound on the rule count), compared to standard grid partitioning. Note that the concept approximates the underlying function through a continuous and linear function. Obviously, the approximation error will increase as the underlying function deviates from being continuous and linear.

5. Approximation analysis

To analyze the FDM approximation mechanism, we initially introduce the following definitions:

Definition 1 (*Normal Fuzzy Set [34]*). A fuzzy set A on a domain U is said to be a normal fuzzy set if there exists at least one $\mathbf{x} \in U$ such that the corresponding membership function returns 1, i.e., $\mu_A(\mathbf{x}) = 1$.

Definition 2 (*Completeness of Partition [34]*). Fuzzy sets A_i where $i = 1, \ldots, h$ form a complete partition on a domain U if for any $\mathbf{x} \in U$ there exists at least one fuzzy set with $\mu_{A_i}(\mathbf{x}) > 0$.

Definition 3 (*Consistency of Partition [34]*). Fuzzy sets A_i where $i = 1, \ldots, h$ form a consistent partition on a domain U if $\mu_{A_i}(\mathbf{x}) = 1$ for any $\mathbf{x} \in U$ and $\mu_{A_j}(\mathbf{x}) = 0$ for all $j \neq i$.

Definition 4 (*Fuzzy Set Support*). The fuzzy support of an input value $\mathbf{x} \in U$ is a union of fuzzy sets A_i of which support includes the given inputs, i.e., $P(\mathbf{x}) = \bigcup_{\mathbf{x} \in A_i} A_i$.

With the above definitions, the approximation mechanism of the FTU (9) is defined by the following theorem:

Theorem 1 (Approximation Mechanism of FTU). *Suppose that fuzzy sets $A^i_{j_i}$ are normal, complete and consistent on training data set S where i = 1, ..., n, j_i = 1, ..., h_i. Note that h_i is the grid count for the ith variable. Then, for any $\mathbf{x} \in I$ and corresponding optimal assignment decision vector $\mathbf{y}^* \in O$, the approximation error is*

$$\epsilon = \|\mathbf{y}^* - FTU(\mathbf{x})\| \tag{28a}$$

$$= \|\mathbf{y}^* - \sum_{j_1,j_2,\ldots,j_n \in H} \phi_{j_1,\ldots,j_n} \mathbf{B}_{\mathbf{j_1},\ldots,\mathbf{j_n}}\| \tag{28b}$$

$$\leqslant \max_{j_1,j_2,\ldots,j_n \in H(\mathbf{x})} \|\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1},\mathbf{j_2},\ldots,\mathbf{j_n}}\| \tag{28c}$$

$$\leqslant \max_{\langle \bar{\mathbf{x}},\bar{\mathbf{y}}\rangle \in S} \{\|\mathbf{y}^* - \bar{\mathbf{y}}\| | \bar{\mathbf{x}} \in P(\mathbf{x})\} \tag{28d}$$

where $A_{j_1,j_2,\ldots,j_n} = A^1_{j_1} \times A^2_{j_2} \times \cdots \times A^n_{j_n}$, $P(\mathbf{x})$ is the fuzzy set support of input $\mathbf{x}$ and $H(\mathbf{x}) = \{j_1,j_2,\ldots,j_n | \mathbf{x} \in A_{j_1,j_2,\ldots,j_n}; j_1,j_2,\ldots,j_n \in H\}$. Note that, $\|\cdot\|$ is the 2-norm or the Euclidean length of the input.

Proof 1. Because $A_{j_1,j_2,\ldots,j_n}$ are complete and consistent fuzzy sets, there exists at least one $A_{j_1,j_2,\ldots,j_n}$ such that $\mathbf{x} \in A_{j_1,j_2,\ldots,j_n}$, where $j_1, j_2, \ldots, j_n \in H$. Recall that $H(\mathbf{x}) = \{j_1,j_2,\ldots,j_n | \mathbf{x} \in A_{j_1,j_2,\ldots,j_n}; j_1,j_2,\ldots,j_n \in H\}$. Therefore,

$$j_1,j_2,\ldots,j_n \in H(\mathbf{x}) \iff \mathbf{x} \in A_{j_1,j_2,\ldots,j_n} \iff \phi_{j_1,j_2,\ldots,j_n}(\mathbf{x}) > 0 \tag{29}$$

However,

$$j_1, j_2, \ldots, j_n \notin H(\mathbf{x}) \iff \mathbf{x} \notin A_{j_1, j_2, \ldots, j_n} \iff \phi_{j_1, j_2, \ldots, j_n}(\mathbf{x}) = 0 \tag{30}$$

In addition, note that

$$\sum_{j_1, \ldots, j_n \in H} \phi_{j_1, \ldots, j_n}(\mathbf{x}) = 1 \tag{31a}$$

$$\sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \varphi_{j_1, \ldots, j_n}(\mathbf{x}) = 1 \tag{31b}$$

Having (30) and (31),

$$\epsilon = \|\mathbf{y}^* - FTU(\mathbf{x})\| \tag{32a}$$

$$= \left\| \mathbf{y}^* - \sum_{j_1, \ldots, j_n \in H} \phi_{j_1, \ldots, j_n}(\mathbf{x}) \mathbf{B}_{\mathbf{j_1}, \ldots, \mathbf{j_n}} \right\| \tag{32b}$$

$$= \left\| \sum_{j_1, \ldots, j_n \in H} \phi_{j_1, \ldots, j_n}(\mathbf{x})(\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1}, \ldots, \mathbf{j_n}}) \right\| \tag{32c}$$

$$= \left\| \sum_{j_1, \ldots, j_n \in H(\mathbf{x})} \phi_{j_1, \ldots, j_n}(\mathbf{x})(\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1}, \ldots, \mathbf{j_n}}) \right\| \tag{32d}$$

$$\leqslant \max_{j_1, \ldots, j_n \in H(\mathbf{x})} \|(\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1}, \ldots, \mathbf{j_n}})\| \tag{32e}$$

Now, simplify the Right Hand Side (RHS) of (32e) as follows:

$$\begin{aligned} &\max_{j_1, \ldots, j_n \in H(\mathbf{x})} \|\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1}, \ldots, \mathbf{j_n}}\| \\ &\quad = \max_{j_1, \ldots, j_n \in H(\mathbf{x})} \left\| \mathbf{y}^* - \sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \varphi_{j_1, \ldots, j_n}(\bar{\mathbf{x}}) \bar{\mathbf{y}} \right\| \\ &\quad = \max_{j_1, \ldots, j_n \in H(\mathbf{x})} \left\| \sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \varphi_{j_1, \ldots, j_n}(\bar{\mathbf{x}})(\mathbf{y}^* - \bar{\mathbf{y}}) \right\| \end{aligned}$$

Note that $\varphi_{j_1, \ldots, j_n}(\bar{\mathbf{x}})$ is non-zero for $\bar{\mathbf{x}} \in A_{j_1, j_2, \ldots, j_n}$. Thus, we may drive the following result.

$$\bar{\mathbf{x}} \in A_{j_1, j_2, \ldots, j_n} \Rightarrow \varphi_{j_1, \ldots, j_n}(\bar{\mathbf{x}}) > 0 \tag{34}$$

Let $j_1, \ldots, j_n \in H(\mathbf{x})$. Thus, we may rewrite (34) as the following:

$$j_1, \ldots, j_n \in H(\mathbf{x}) \wedge \bar{\mathbf{x}} \in P(\mathbf{x}) \Rightarrow \varphi_{j_1, \ldots, j_n}(\bar{\mathbf{x}}) > 0 \tag{35}$$

Having (35), we may further simplify the RHS of (32e) as follows:

$$RHS \leqslant \max_{\bar{\mathbf{x}} \in P(\mathbf{x})} \|\mathbf{y}^* - \bar{\mathbf{y}}\| \quad \Box \tag{36}$$

Theorem 1 introduces a local approximation property of the FTU in a compact form. We extend this local property to a global one, thus setting a bound on the FTU output by the following theorem:

Theorem 2 (Approximation Bound of FTU). *Suppose that fuzzy sets $A^i_{j_i}$ of the FTU (6) are normal, complete and consistent on the data set S.*

i. *Let*

$$\epsilon_{j_1, \ldots, j_n} = \sup_{\mathbf{x} \in A_{j_1, \ldots, j_n}} \|\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1}, \ldots, \mathbf{j_n}}\| \tag{37}$$

be the approximation error of any FTU rule (4) for any arbitrary input enclosed by the rule premise, i.e., $\mathbf{x} \in A_{j_1, \ldots, j_n}$, where $\mathbf{y}^* \in O$ is the corresponding optimal assignment decision vector and i = 1, . . . , n, j_i = 1, . . . , h_i. Note that h_i is the grid count for the ith variable. The approximation bound is then

$$\mathrm{BOUND}_1 = \sup_{\mathbf{x}\in I}\|\mathbf{y}^* - FTU(\mathbf{x})\| \tag{38a}$$
$$= \max_{j_1,\ldots,j_n\in H}\{\epsilon_{j_1,\ldots,j_n}\} \tag{38b}$$

ii. *Let*

$$\rho_{\langle\bar{\mathbf{x}},\bar{\mathbf{y}}\rangle} = \sup_{\mathbf{x}\in P(\bar{\mathbf{x}})}\|\mathbf{y}^* - \bar{\mathbf{y}}\| \tag{39}$$

be the approximation error introduced by the training data $\langle\bar{\mathbf{x}},\bar{\mathbf{y}}\rangle \in S$. *The approximation bound is then*

$$\mathrm{BOUND}_2 = \sup_{\mathbf{x}\in I}\|\mathbf{y}^* - FTU(\mathbf{x})\| \tag{40a}$$
$$= \max_{\langle\bar{\mathbf{x}},\bar{\mathbf{y}}\rangle\in S}\{\rho_{\langle\bar{\mathbf{x}},\bar{\mathbf{y}}\rangle}\} \tag{40b}$$

iii.

$$\mathrm{BOUND}_1 \leqslant \mathrm{BOUND}_2 \tag{41}$$

Note that $\|\cdot\|$ *is the 2-norm or the Euclidean length of the input.*

Proof 2. Remember the Theorem 1 defining the FTU's approximation mechanism for any input and corresponding decision, i.e., $\mathbf{x}\in I$ and $\mathbf{y}^*\in O$:

$$\epsilon = \|\mathbf{y}^* - FTU(\mathbf{x})\| \tag{42a}$$
$$\leqslant \max_{j_1,\ldots,j_n\in H(\mathbf{x})}\|\mathbf{y}^* - \mathbf{B}_{\mathbf{j_1},\ldots,\mathbf{j_n}}\| \tag{42b}$$
$$\leqslant \max_{\bar{\mathbf{x}}\in P(\mathbf{x})}\|\mathbf{y}^* - \bar{\mathbf{y}}\| \tag{42c}$$

where $\langle\bar{\mathbf{x}},\bar{\mathbf{y}}\rangle \in S$.

(i) Note that we may write the following fact for any $\mathbf{x}\in A_{j_1,\ldots,j_n}$:

$$\|\mathbf{y}^* - \mathbf{B}_{\mathbf{h_1},\ldots,\mathbf{h_n}}\| \leqslant \sup_{\mathbf{x}^*\in A_{j_1,\ldots,j_n}}\|\mathbf{y}^{**} - \mathbf{B}_{\mathbf{j_1},\ldots,\mathbf{j_n}}\| \tag{43}$$

where $\mathbf{y}^*$ and $\mathbf{y}^{**}$ are the optimal assignments corresponding to the inputs $\mathbf{x}$ and $\mathbf{x}^*$, respectively. Having (43), we may extend (42b) to a global bound as follows:

$$\begin{aligned}
BOUND &= \sup_{\mathbf{x}\in I}\|\mathbf{y}^* - FTU(\mathbf{x})\| \\
&\leqslant \sup_{\mathbf{x}\in I}\left\{\max_{h_1,\ldots,h_n\in H(\mathbf{x})}\|\mathbf{y}^* - \mathbf{B}_{\mathbf{h_1},\ldots,\mathbf{h_n}}\|\right\} \\
&\leqslant \sup_{\mathbf{x}\in I}\left\{\max_{h_1,\ldots,h_n\in H(\mathbf{x})}\ldots\right. \\
&\quad\left.\ldots\left\{\sup_{\mathbf{x}^*\in A_{j_1,\ldots,j_n}}\|\mathbf{y}^{**} - \mathbf{B}_{\mathbf{j_1},\ldots,\mathbf{j_n}}\|\right\}\right\} \\
&\leqslant \max_{h_1,\ldots,h_n\in H}\left\{\sup_{\mathbf{x}^*\in A_{j_1,\ldots,j_n}}\|\mathbf{y}^{**} - \mathbf{B}_{\mathbf{j_1},\ldots,\mathbf{j_n}}\|\right\}
\end{aligned}$$

(ii) For any $\mathbf{x}\in P(\bar{\mathbf{x}})$, we can obtain the following result:

$$\|\mathbf{y}^* - \bar{\mathbf{y}}\| \leqslant \sup_{\mathbf{x}^*\in P(\bar{\mathbf{x}})}\|\mathbf{y}^{**} - \bar{\mathbf{y}}\| \tag{44}$$

where $\mathbf{y}^*$ and $\mathbf{y}^{**}$ are the optimal assignments corresponding to the inputs $\mathbf{x}$ and $\mathbf{x}^*$, respectively. Having (44), we may extend (42c) to a global bound as follows:

$$\begin{aligned}
BOUND &= \sup_{\mathbf{x}\in I}\|\mathbf{y}^* - FTU(\mathbf{x})\| \\
&\leqslant \sup_{\mathbf{x}\in I}\left\{\max_{\bar{\mathbf{x}}\in P(\mathbf{x})}\|\mathbf{y}^* - \bar{\mathbf{y}}\|\right\} \\
&\leqslant \sup_{\mathbf{x}\in I}\left\{\max_{\bar{\mathbf{x}}\in P(\mathbf{x})}\sup_{\mathbf{x}^*\in P(\bar{\mathbf{x}})}\|\mathbf{y}^{**} - \bar{\mathbf{y}}\|\right\} \\
&\leqslant \max_{\bar{\mathbf{x}}\in S}\left\{\sup_{\mathbf{x}^*\in P(\bar{\mathbf{x}})}\|\mathbf{y}^{**} - \bar{\mathbf{y}}\|\right\}
\end{aligned}$$

where $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S$.
(iii) is obvious by the (42). □

We have analyzed the approximation mechanism of the FTU, and can now check the approximation accuracy of the FTU. Note that the FTU output does not necessarily constitute a valid decision vector. FCUs map the FTU output to a valid decision vector. Therefore, we must also set an optimality bound for the FCUs (17) that is also the optimality bound of the FDM. To do so, we compare the FCU output with the optimal decision vector through the cost function associated with the minimization problem (1) as follows:

Theorem 3 (*Optimality Bound of the FDM*). *Suppose that the fuzzy sets $C^i_{j_i}$ ($i = 1, \ldots, m$ and $j_i = 1, \ldots, g_i$) are normal, complete and consistent on the data set S' (13). Let $COST(\cdot)$ be the cost function associated with the minimization function (1). Let*

$$\epsilon_{j_1,\ldots,j_m} = \sup_{\mathbf{u} \in C_{j_1,\ldots,j_m}} |COST(\mathbf{y}^*) - COST(\mathbf{D_{j_1,\ldots,j_m}})| \tag{45}$$

be the approximation error for any arbitrary FCU layer input mapped to the fuzzy set $C_{j_1 j_2,\ldots,j_m}$, which is $\mathbf{u} = FTU(\mathbf{x}) \in C_{j_1 j_2,\ldots,j_m}$ where $C_{j_1 j_2,\ldots,j_m} = C^1_{j_1} \times C^2_{j_2} \times \ldots. \times C^m_{j_m}$. Note that $\mathbf{y}^* \in O$ is the optimal assignment decision vector for the input $\mathbf{x} \in I$. The optimality bound is then s

$$\begin{aligned} \text{BOUND}_3 &= \sup_{\mathbf{x} \in I} |COST(\mathbf{y}^*) - COST(FDM(\mathbf{x}))| \\ &= \max_{j_1,\ldots,j_m \in G} \{\epsilon_{j_1,\ldots,j_m}\} \end{aligned} \tag{46a}$$

where $G = \{j_1, j_2, \ldots, j_m | j_1 = 1, 2, \ldots, g_1; j_2 = 1, 2, \ldots, g_2, \ldots, j_m = 1, 2, \ldots, g_m\}$

Proof 3. Because $C_{j_1 j_2,\ldots,j_m}$ are complete and consistent, there exists at least one $C_{j_1 j_2,\ldots,j_m}$ such that $\mathbf{u} = FTU(\mathbf{x}) \in C_{j_1 j_2,\ldots,j_m}$ for any $\mathbf{x} \in I$ where $j_1, j_2, \ldots, j_m \in G$. Moreover, because the FCU model (17) is a typical fuzzy classifier, the output of the FCU corresponds to exactly one of the possible assignment decisions. Thus, we gain the following result: $\mathbf{y} = FCU(\mathbf{u}) \in \{\mathbf{D_{j_1,\ldots,j_m}} | j_1, \ldots, j_m \in G(\mathbf{u})\}$ where $G(\mathbf{u}) = \{j_1, \ldots, j_m | \mathbf{u} \in C_{j_1,\ldots,j_m}; j_1, \ldots, j_m \in G\}$. We may use following inequality as the approximation mechanism of the FDM, as defined by (6) and (17):

$$|COST(\mathbf{y}^*) - COST(FDM(\mathbf{x}))| \leqslant \max_{j_1,\ldots,j_m \in G(\mathbf{u}=FTU(\mathbf{x}))} |COST(\mathbf{y}^*) - COST(\mathbf{D_{j_1,\ldots,j_m}})| \tag{47}$$

for any $\mathbf{x} \in I$ where $FDM(\mathbf{x}) = FCU(FTU(\mathbf{x}))$. Note that $\mathbf{y}^*$ is the optimal assignment corresponding to the input $\mathbf{x}$. From (47), the global bound for the FDM will be

$$\begin{aligned} BOUND = \sup_{\mathbf{x} \in I} |COST(\mathbf{y}^*) - COST(FDM(\mathbf{x}))| &\leqslant \sup_{\mathbf{x} \in I} \left\{ \max_{j_1,\ldots,j_m \in G(\mathbf{u}=FTM(\mathbf{x}))} \ldots |COST(\mathbf{y}^*) - COST(\mathbf{D_{j_1,\ldots,j_m}})| \right\} \\ &\leqslant \max_{j_1,\ldots,j_m \in G} |COST(\mathbf{y}^*) - COST(\mathbf{D_{j_1,\ldots,j_m}})| \quad \square \end{aligned}$$

Theorems 2 and 3 allow for checking the approximation accuracy on a given data set. The next issue is guaranteeing the optimality of the FDM. Note that the optimal decisions are all in the approximation bound neighborhood of at least one of the fuzzy sets belonging to the FCUs. As an illustrative example, fuzzy set $C_{j_1 j_2,\ldots,j_m}$ is shown in Fig. 2 with $\mathbf{u_1}$, $\mathbf{u_2}$ and $\mathbf{u_3}$, which are any input for the FCUs and also any output of the FTU. Let $\mathbf{y}^*_1, \mathbf{y}^*_2$, and $\mathbf{y}^*_3$ be the optimal assignments corresponding to these inputs/outputs. As seen from Fig. 2, the optimal assignments are in the approximation bound neighborhood of $C_{j_1 j_2,\ldots,j_m}$. Note that the FCUs map any input falling in $C_{j_1 j_2,\ldots,j_m}$, i.e., $\mathbf{u} \in C_{j_1,\ldots,j_m}$, to $\mathbf{D_{j_1,\ldots,j_m}}$. It is obvious that this mapping is optimal if the optimal assignment for any input falling in the fuzzy set $C_{j_1 j_2,\ldots,j_m}$ is exactly equal to $\mathbf{D_{j_1,\ldots,j_m}}$ for all $j_1, j_2, \ldots, j_m \in G$.

Based on the rationale provided above, we can guarantee the optimality of the FDM as follows: (i) Originate rule-bases through normal, complete and consistent fuzzy sets for both the FCU and FTU. (ii) Any fuzzy set belonging to FCUs (i.e., $C_{j_1 j_2,\ldots,j_m}$ for any $j_1, \ldots, j_m \in G$) must include exactly only one valid decision. (iii) The approximation bound of the FTU must be small enough to ensure that there exists one fuzzy set in the FCU layer that includes both the FTU output and its corre-

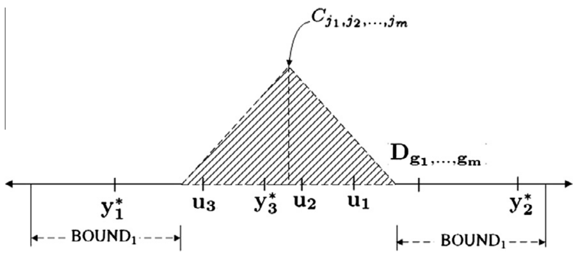

Fig. 2. The fuzzy set, $C_{j_1 j_2,\ldots,j_m}$, on the variable axis.

sponding optimal decision. We assert that these three statements guarantee an optimally trained FDM for a given data set. Note that it is easy to satisfy statement (ii) with small rule-bases for the FCU layer because it requires only $(t+1)^w$ rules. However, statement (iii) coexists with complex FTU models, denoting large rule-bases.

6. Simulations and results

In this section, we analyze and discuss the performance of the FDM and the effectiveness of the proposed grid partitioning methodology. Because the primary goal of this work is to show the applicability of fuzzy decision making in weapon target assignment, we use small-scale scenarios to test the approach (i.e., a single platform perspective). To do so, we consider two different small-scale illustrative examples with two and four weapons. As a third example, a large-scale scenario with twenty weapons and forty targets is considered to show the scalability of the approach. Note that this work primarily concentrates on small-scale WTA instances because the goal is to show the possibility of applying approximate reasoning theory to WTA. To test how FDMs perform in practice, we apply a K-fold cross validation [17]. In K-fold cross validation, the original sample set is randomly partitioned into k subsets. $k-1$ subsets are used in training, and the remaining subset is set aside for validation. The cross-validation process is then repeated k times such that each subset is used once as the validation set. Thus, all subsets are used both for training and validation. We implement the models and perform all of our tests with MATLAB 7.0 on a PC with a 2.4 GHz P4 processor and 3 GB of RAM.

Table 1
Analysis results of the FDM models for the scenario 1.

Solver	Theoretical		Average of simulations		
	Approx. bound	Optim. bound	Approx. error	Optim. error	Comp. time (s)
Fuzzy Decision Maker (FDM) Standard Grid Partitioning Rule Count (FTU): 512 Rule Count (FCU Layer): 54 Triangular membership functions	0.97	0.57	0.74 (0.12)	0.05 (0.008)	<0.01 (≈0.0)
Fuzzy Decision Maker (FDM) Standard Grid Partitioning Rule Count (FTU): 512 Rule Count (FCU Layer): 54 Trapezoid membership functions	0.97	0.46	0.75 (0.12)	0.04 (0.007)	<0.01 (≈ 0.0)
Fuzzy Decision Maker (FDM) Standard Grid Partitioning Rule Count (FTU): 19,683 Rule Count (FCU Layer): 686 Triangular membership functions	0.71	0.31	0.60 (0.07)	0.01 (0.004)	0.028 (≈0.0)
Fuzzy Decision Maker (FDM) Standard Grid Partitioning Rule Count (FTU): 19,683 Rule Count (FCU Layer): 686 Trapezoid membership functions	0.63	0.27	0.54 (0.07)	0.01 (0.003)	0.028 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level Rule Count (FTU): 125 Rule Count (FCU Layer): 48 Trapezoid membership functions	0.68	0.81	0.62 (0.04)	0.05 (0.013)	<0.01 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level Rule Count (FTU): 2352 Rule Count (FCU Layer): 91 Trapezoid membership functions	0.55	0.21	0.47 (0.03)	0.01 (0.005)	<0.01 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level One rule per valid assignment decision Rule Count (FTU): 108,864 Rule Count (FCU Layer): 8 Trapezoid membership functions	0.03	0.0	< 0.01 (≈ 0.0)	0.0 (0.0)	0.078 (≈0.0)
Branch and Bound Algorithm (BBA) [1]	–	–	–	0.0 (0.0)	0.46 (0.28)
Genetic Algorithm (GA) [20]	–	–	–	0.00 (0.0)	0.2 (≈0.0)

6.1. Scenario 1: $w = 2$ and $t = 3$

We define a simple small-scale scenario involving two weapons and three targets. We choose an illustrative function to evaluate the target value as follows:

$$v_i = 1 - r_i/rmax \tag{48}$$

where $r_i \in [0, rmax]$ is the range of the target $i = 1, \ldots, t$ and $rmax$ is the maximum scenario range. We choose a Gaussian function to characterize the weapon kill probability as follows:

$$p_{ij} = a_j e^{-\frac{(r_i - b_j)^2}{c_j^2}} \tag{49}$$

where a_j is the maximum kill probability for weapon $j = 1, \ldots, w$, b_j is the range where weapon j has the maximum kill probability, and c_j is the width of the function bell (i.e., standard deviation). For this scenario, we set the maximum kill probabilities as $a_1 = 0.7$ and $a_2 = 0.4$, the ranges as $b_1 = 5000$ and $b_2 = 20{,}000$, and the standard deviations as $c_1 = 5000$ and $c_2 = 20{,}000$.

We sample 10^4 instances such that each instance corresponds to an arbitrary vector of target ranges. For each instance, we evaluate the target values and the weapon kill probabilities by means of (48) and (49). We then calculate the exact decisions d_{ij} using a branch and bound algorithm for a particular v_i and p_{ij}. Next, we generate data tuples $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle$ by (2) and (3), finally forming training data set S. To elaborate the principals of the approach, we train several FDMs by varying the training parameters the rule generation method, the size of the rule-base, and the membership function. The fuzzy sets are all normal, complete and consistent. In addition, we apply the method described in Section 5 to guarantee the optimality of the FDM. We also implement the branch and bound algorithm (BBA) [1] and the genetic algorithm (GA) [20] for comparison purposes. Thus, the performance of the exact solver, the suboptimal solver and the FDM can be compared according to both computation time and optimization error. To conduct a fair comparison, the parameters of BBA and GA are taken from [1,20].

To evaluate the performance of the FDM, we use both theoretical and simulated results. For the theoretical analysis, we compute the approximation and optimality bounds by means of Theorems 2 and 3 and report the results in Table 1 (the 'Approx. Bound' and 'Optim. Bound' columns). We then directly compute the approximation and optimality errors within the simulations, where we compare the FDM output (i.e., $\mathbf{y}$) with the optimal assignment (i.e., $\mathbf{y}^*$). To evaluate how the FDM performs in practice, we apply a 10-fold cross validation. The approximation error is calculated by taking the norm (2-norm) of the difference between the optimal assignment and the FTU output, i.e., $\|\mathbf{y}^* - \mathbf{u}\|$ where $\mathbf{u} = FTU(\mathbf{x})$. The optimality error is calculated by taking the difference between the cost of the optimal assignment and the cost of the FDM output and normalizing the difference by the cost of the optimal assignment. That is, $(COST(\mathbf{y}) - COST(\mathbf{y}^*))/COST(\mathbf{y}^*)$, where $\mathbf{y} = FCU(FTU(\mathbf{x}))$. Note that $COST(\cdot)$ is the cost function associated with the minimization problem (1). We report the simulation results in Table 1 (the 'Approx. Error' and the 'Optim. Error' columns). In addition, standard deviations are provided in parentheses where applicable.

Table 2
Analysis results of the FDM models for the scenario 2.

Solver	Theoretical		Average of Simulations		
	Approx. bound	Optim. bound	Approx. error	Optim. error	Comp. time (s)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level Rule Count (FTU): 1260 Rule Count (FCU Layer): 350 Trapezoid membership functions	2.03	0.39	1.68 (0.11)	0.26 (0.052)	<0.01 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level Rule Count (FTU): 110,592 Rule Count (FCU Layer): 732 Trapezoid membership functions	1.83	0.14	1.35 (0.09)	0.03 (0.006)	0.11 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level One rule per valid assignment decision Rule Count (FTU): 663,552 Rule Count (FCU Layer): 28 Trapezoid membership functions	0.07	0.0	<0.01 (≈0.0)	<0.001 (≈0.0)	0.48 (≈0.0)
Branch and Bound Algorithm (BBA) [1]	–	–	–	0.0 (0.0)	13.56 (7.4)
Genetic Algorithm (GA) [20]	–	–	–	0.0 (0.0)	0.7 (≈0.1)

6.2. Scenario 2: $w = 4$ and $t = 6$

As the second illustrative example, we define another small-scale scenario involving 4 weapons and 6 targets. Similar to the previous example, we characterize the target values and the weapon kill probabilities by (48) and (49). We set $a_1 = 0.7$, $a_2 = 0.8$, $a_3 = 0.8$ and $a_4 = 0.99$; $b_1 = 50{,}000$, $b_2 = 15{,}000$, $b_3 = 15{,}000$ and $b_4 = 5000$; $c_1 = 3000$, $c_2 = 5000$, $c_3 = 5000$ and $c_4 = 1000$. In defining the scenario, we form a data set with 10^4 data tuples.

With this data set, we train the FDMs in a similar manner to the previous example. We report the theoretical and simulated results in Table 2.

6.3. Scenario 3: $w = 10$ and $t = 10$

As the third illustrative example, we define a larger scenario to demonstrate the scalability of the approach. In this case, the weapon count is 10 and the target count is (at most) 10. We characterize the target values and the weapon kill probabilities using (48) and (49). We set $a_j = 0.7$ for $j = 1, 2, 3$, $a_j = 0.8$ for $j = 4, 5, 6$, $a_j = 0.8$ for $j = 7, 8$, $a_j = 0.99$ for $j = 9, 10$; $b_j = 50{,}000$ for $j = 1, 2, 3$, $b_j = 15{,}000$ for $j = 4, 5, 6$, $b_j = 15{,}000$ for $j = 7, 8$, $b_j = 5000$ for $j = 9, 10$; $c_j = 3000$ for $j = 1, 2, 3$, $c_j = 5000$ for $j = 4, 5, 6$, $c_j = 5000$ for $j = 7, 8$, $c_j = 1000$ for $j = 9, 10$. In defining the scenario, we form a data set with 10^4 data tuples.

With this data set, we train the FDMs in a manner similar to the previous examples. We report the theoretical and simulated results in Table 3.

6.4. Discussion of results

Tables 1–3 clearly show that the theoretical results agree with simulated results (in approximation and optimality errors). The approximation bound is slightly greater than the approximation error in each case. However, the optimality bound is significantly larger than the optimality error. These results show that the theoretical analysis (of the approximation and optimality bounds) is valid. Moreover, we can state that while the approximation bound is strong, the optimality bound is weak.

We can train optimal FDMsas seen from Tables 1–3 by applying the method described in Section 5. Therefore, this method appears to be valid. However, it forces the FTUs to use complex rule-bases, resulting in excessive memory and an increase in computation time. We note that the optimality can only be guaranteed for the training data set. It is not expected that the optimality error is zero for any simulation because we apply a K-fold cross validation.

The simulation results in Tables 1 and 2 show that FDMs with moderate or large rule-bases have satisfactory approximation performances compared to BBAs and GAs. Furthermore, the FDMs are more efficient in computation time compared to the other solvers. Because it is possible to simultaneously evaluate (or run) rules (i.e., concurrently) for fuzzy systems, the total computation time is theoretically equal to the computation time of a single rule. Moreover, it is highly likely that the computation time would be worse for BBAs and Gas, as the cost calculation function becomes more complex because these

Table 3
Analysis results of the FDM models for scenario 3.

Solver	Theoretical		Average of Simulations		
	Approx. bound	Optim. bound error	Approx. error	Optim.	Comp. time (s)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level Rule Count (FTU): 419,904 Rule Count (FCU Layer): 12,230 Trapezoid membership functions	6.23	0.69	3.54 (0.22)	0.30 (0.05)	0.30 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level Rule Count (FTU): 6,336,000 Rule Count (FCU Layer): 51,159 Trapezoid membership functions	4.20	0.27	2.85 (0.22)	0.01 (0.006)	4.53 (≈0.0)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level One rule per valid assignment decision Rule Count (FTU): 707,788,800 Rule Count (FCU Layer): 110 Trapezoid membership functions	0.06	0.0	<0.01 (≈0.0)	<0.001 (≈0.0)	512 (≈0.0)
Branch and Bound Algorithm (BBA) [1]	–	–	–	0.0 (0.0)	399 (204)
Genetic Algorithm (GA) [20]	–	–	–	0.0 (0.0)	3.2 (0.4)

search-based solvers run cost functions for each enumeration (candid decision). The complexity of the cost function does not affect the performance of the FDM. It should be noted that the solvers are all implemented in MATLAB on a PC; it is quite possible to considerably decrease the computation time by implementing the solvers on more sophisticated platforms with more sophisticated programming tools. Even in this case, the FDM still would be more efficient in terms of computation time due to its concurrent processing capability and its independence from the cost function. In addition, we notice from Table 1 that the membership function type has only a slight effect on approximation and optimization errors. Therefore, the approximation performance of the FDMs can be slightly increased by optimizing the membership functions.

Based on Table 1, it is evident that the FDMs trained by the sensitivity-based grid partitioning can achieve the desired performance with smaller rule-bases compared to the FDMs trained by the standard grid partitioning. Therefore, less memory is required in the hardware to implement the FDM with sufficient accuracy.

Table 3 shows that the FDMs exhibit satisfactory approximation performances with complex (large) rule-bases. Although the FDM has concurrent processing capabilities, it still must sum a large number of outputs (rule outputs) for a complex rule-base, thus considerably increasing the computation time. The FDM appears to be scalable (to some extent) for WTA. However, the simulation results above show that FDMs are quite satisfactory for small-scale WTAs where both weapon and threat counts are small. The FDM can clearly be used as a decision aid system to help commanders for small scale WTA.

7. Conclusions

In this work, we propose an alternative decision aid system for WTA to assist commanders on the battlefield. The concept defines a rule-based mapping through approximate reasoning theory. This work is the first application of approximate reasoning theory to WTA by applying a grid partitioning approach to generate rule-bases. In addition to grid partitioning, we propose grid counts by considering the sensitivity level of each input variable. Moreover, we suggest a method to guarantee the optimality. The trained FDMs show satisfactory performance, particularly for small-scale WTA.

We rely on other solvers to provide samples while training the FDMs, meaning that once the problem size is changed, we must reform the training data set. To shorten the computation time required to form the training data set on the battlefield, we propose two robust alternatives. The first (and obvious) alternative is to form several training sets for each possible scenario and train one FDM for each training set. Once the problem size is changed for the first alternative, one simply uses the appropriately trained FDM. The first alternative satisfies the optimality but may require a large number of stored FDMs. The second alternative is to train only one FDM (considering the maximum possible assignments) such that we form a single training data set where the weapon count is exactly equal to the number of weapons on the platform (and the target count is set equal to the weapon count). The second alternative is applied as follows: When the real target count (denoted as t_r) is greater than the target count of the training set (denoted as t), we sort the real targets according to the target values and run the FDM for the first t targets. When the real target count is less than the target count of the training set, we create $t - t_r$ artificial targets with zero values and run the FDMs for t_r real targets and $t - t_r$ artificial targets. Though the second alternative does not require the storage of a large number of FDMs, it does not satisfy the global optimality.

The approach appears to be scalable to some extent. We believe that the reason for this limitation is the grid partitioning method. Suppose that an assignment problem is to be solved for w weapons against maximum t targets. When we divide each input variable into h fuzzy grids, a complete rule-base should have h^n fuzzy rules where $n = t + wt$. Obviously, it is infeasible to design a fuzzy decision maker for weapon target assignment with a large number of input variables. In our work, we address this problem by choosing the grid counts, considering the sensitivity level of each input variable. However, there are a large number of more sophisticated rule extraction methods in the existing literature. Future works could apply state-of-the-art rule extraction methods within our approach to reduce the number of rules, thus saving computation time.

We should note that the sensitivity level is computed on a given data set by approximating the underlying function to a linear continuous function. We believe that this idea can be further extended by choosing other functions in which the underlying function is approximated. Moreover, to guarantee the optimality in this work, we require complex FTUs (i.e., large rule-bases). We believe that the theoretical study could be improved to achieve better approximation and optimality bounds, which may lead to new methods to guarantee the optimality.

Acknowledgement

Authors would like to thank HAVELSAN for its support.

References

[1] R.K. Ahuja, A. Kumar, K. Jha, J.B. Orlin, Exact and Heuristic Methods for the Weapon Target Assignment Problem (Working Paper: 4464-03), MIT Sloan School of Management, Cambridge, MA, USA, 2003.
[2] R.K. Ahuja, A. Kumar, K. Jha, J.B. Orlin, Exact and heuristic algorithms for the weapon-target assignment problem, Operations Research 55 (6) (2007) 1136–1146.
[3] A.E. Bayrak, F. Polat, Employment of an evolutionary heuristic to solve the target allocation problem efficiently, Information Sciences 222 (2013) 675–695.
[4] D. Blodgett, M. Gendreau, F. Guertin, J.Y. Potvin, A tabu search heuristic for resource management in naval warfare, Journal of Heuristics 9 (2003) 145–169.

[5] Z. Bogdanowicz, A new efficient algorithm for optimal assignment of smart weapons to targets, Computers and Mathematics with Applications 58 (10) (2009) 169–174.
[6] I. Boussaid, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Information Sciences 237 (2013) 82–117.
[7] R.E. Burkard, F.A. Rendl, A thermodynamically motivated simulation procedure for combinatorial optimization problems, European Journal of Operational Research 17 (2) (1984) 169–174.
[8] H. Cai, J. Liu, Y. Chen, H. Wang, Survey of the research on dynamic weapon-target assignment problem, Journal of Systems Engineering and Electronics 17 (3) (2006) 559–565.
[9] S.C. Chang, R.M. James, J.J. Shaw, Assignment algorithm for kinetic energy weapons in boost defence, in: Proceedings of IEEE Decision and Control Conference, Los Angeles, CA, USA, 1987.
[10] J. Chen, B. Xin, Z. Peng, L. Dou, J. Zhang, Evolutionary decision-makings for the dynamic weapon-target assignment problem, Science in China Series F: Information Sciences 52 (11) (2009) 2006–2018.
[11] R.H. Day, Allocating weapons to target complexes by means of non-linear programming, Operations Research 14 (1966) 992–1013.
[12] G.G. DenBroeder, R.E. Ellison, L. Emerling, On optimum target assignments, Operations Research 7 (3) (1959) 322–326.
[13] A.R. Eckler, S.A. Burr, Mathematical Models of Target Coverage and Missile Allocation (Technical Report DTIC: AD-A953517), Military Operations Research Society, Alexandria, USA, 1972.
[14] S. Guillaume, Designing fuzzy inference Systems from data: an interpretability-oriented review, IEEE Transactions on Fuzzy Systems 9 (3) (2001) 426–443.
[15] O. Karasakal, N.E. Ozdemirel, L. Kandiller, Anti-ship missile defence of a naval task group, Naval Research Logistics 58 (3) (2011) 304–321.
[16] J.D. Katter, A Solution of the Multi-Weapon, Multi-Target Assignment Problem (Working Paper: 26957), MITRE, Bedford, MA, USA, 1986.
[17] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of IJCAI, Montreal, Canada, 1995, pp. 1137–1143.
[18] M. Lee, Constrained weapon target assignment: enhanced very large scale neighbourhood search algorithm, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 40 (1) (2010) 198–204.
[19] Z.J. Lee, C.Y. Lee, A hybrid search algorithm with heuristics for resource allocation problem, Information Sciences 173 (2005) 155–167.
[20] Z.J. Lee, S.F. Su, C.Y. Lee, Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics, IEEE Transactions on Systems, Man and Cybernetics, Part B 33 (1) (2003) 113–121.
[21] S.P. Lloyd and H.S. Witsenhausen, Weapon allocation is NP-complete, in: Proceedings of Summer Simulation Conference, Reno, NV, USA, 1986, pp. 1054–1058.
[22] A.S. Manne, A target assignment problem, Operations Research 6 (3) (1958) 346–351.
[23] R.A. Murphey, Target-based weapon target assignment problems, in: P.M. Pardalos, L.S. Pitsoulis (Eds.), Non-linear Assignment Problems: Algorithms and Applications, vol. 7, Kluwer Academic Publisher, Dordrecht, 1999, pp. 39–53.
[24] S. Paradis, A. Benaskeur, M. Oxenham, P. Cutler, Threat evaluation and weapons allocation in network-centric warfare, in: Proceedings of Information Fusion, Philadelphia, PA, USA, 2005.
[25] J.M. Rosenberger, A. Yucel, The generalized weapon target assignment problem, in: Command and Control Research and Technical Symposium, McLean, VA, USA, 2005.
[26] J.N. Roux, J.H. Van Vuuren, Threat evaluation and weapon assignment decision support: a review of the state-of-the-art, ORiON: Journal of ORSSA 23 (2) (2007) 151–187.
[27] M.A. Sahin, K. Leblebicioglu, A standard expert system for weapon target assignment problem, in: Proceedings of SPECTS, Istanbul, Turkey, 2009.
[28] M.A. Sahin, K. Leblebicioglu, Rule based weapon target assignment on the battlefield, in: Proceedings of 18th IFAC World Congress, Milano, Italy, 2011.
[29] G.A.F. Seber, C.J. Wild, Nonlinear Regression, Wiley, New York, USA, 1989.
[30] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modelling and control, IEEE Transactions on Systems, Man, Cybernetics 15 (1985) 116–132.
[31] B. Xin, J. Chen, Z. Peng, L. Dou, J. Zhang, An efficient rule-based constructive heuristic to solve dynamic weapon target assignment problem, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41 (3) (2011) 598–606.
[32] B. Xin, J. Chen, J. Zhang, L. Dou, Z. Peng, Efficient decision making for dynamic weapon-target assignment by virtual permutation and tabu search heuristics, IEEE Transactions on Systems, Man and Cybernetics, Part C 40 (6) (2010) 649–662.
[33] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man and Cybernetics 3 (1) (1973) 28–44.
[34] X. Zeng, M. Zeng, Approximation theory of fuzzy systems-MIMO case, IEEE Transactions on Fuzzy Systems 3 (2) (1993) 219–235.
[35] X. Zeng, Y. Zhu, L. Nan, K. Hu, B. Niu, X. He, Solving weapon-target assignment problem using discrete particle swarm optimization, in: Proceedings of Intelligent Control and Automation Conference, Dalian, China, 2006, pp. 3562–3565.
[36] L. Zenghua and W. Jingye, Weapon target assignment research based on genetic algorithm mixed with damage simulation, in: Proceedings of ICCASM, Taiyuan, China, 2010, pp. 460–463.